Enforcing Consistency on Coreference Sets

Klenner, M

Enforcing Consistency on Coreference Sets

Manfred Klenner
Institute of Computational Linguistics
University of Zurich
klenner@cl.uzh.ch

Abstract

We show that intra-sentential binding constraints can act as global constraints that - via transitivity - enforce consistency on coreference sets. This yields about 5 % increase in performance. In our model, the probabilities of a baseline classifier for coreference resolution are used as weights in an optimization model. The underlying integer linear programming (ILP) specification is straightforward - binding constraints give rise to (local) exclusiveness of markables, transitivity propagates these restrictions and enforces the re-computation of coreference sets.

Keywords

Coreference Resolution, Global Constraints, Optimization, Intra-sentential Binding Constraints

1 Introduction

In many NLP fields including coreference resolution, approaches are still striving to improve empirical results by a rather traditional (after about twenty years one might use this term) machine learning system design: given some annotated data for the problem to be solved, find appropriate features and train a classifier, e.g. maximum entropy, decision trees or k-nearest neighbor. Actually, these attempts are successful, there is still room for improvements, for example with respect to coreference resolution through the integration of semantic knowledge from new resources such as Wikipedia [14]. One problem with these approaches, however, is that they can't adhere to - globally operative - prescriptive knowledge. Such strong linguistic principles exist and some of them are never violated even in real scenarios. Take intra-sentential binding constraints as given in the following example:

A [man] stole/sold [him] [his] car.
[Peter] was angry/happy.

It is known from binding theory that "man" as the subject and "him" as the indirect object of the same (non-predicative) verb must have exclusive referents. On the other hand, "his" can be coreferent with either, depending on the verb and world knowledge: "his" refers to "him" in the case of "steal", "his" refers to "man" in the case of "sell"[1].

Assume a binary classifier that incorporates binding constraints in form of a hard filter. This would prevent exclusive pairs such a (man him) from being generated. Unfortunately, such local decisions do not impose any restrictions on the resolution of subsequent pairs: (man his), (him his), (man peter), (him peter), (his peter). However, only some combinations form a consistent solution. For example, {(man his), (him his)} does not, since, via transitivity of the anaphoric relation, (man him) deductively follows. An inconsistent coreference set, thus, evolves. Transitivity of the anaphoric relation is a constraint that cannot be integrated in a binary classifier since it classifies candidate pairs independently of each other.

The crucial point is that a local constraint (intra-sentential binding constraint) becomes via transitivity of the anaphoric relation globally operative. Most of the time no simple repair mechanism operating on the inconsistent classifier output could do a good job, since these dependencies can get rather complex and often there is no single but multiple consistent solutions. The questions is, which is the optimal solution.

We introduce a model of coreference resolution that bases its decisions on the output (probabilities) of a traditional classifier. Our model is formalized within the framework of Integer Linear Programming, a constraint-based numerical optimization algorithm. As long as no inconsistencies arise, the decisions of the baseline classifier are left unaltered. Violations of exclusiveness restrictions as indicated by binding constraints cause a reordering of coreference sets. Our architecture, thus, combines theory-based, apriori linguistic knowledge of the problem at hand with empirically derived preferences.

2 Integer Linear Programming

Integer Linear Programming (ILP)[11] is the name of a class of constraint satisfaction algorithms which are restricted to a numerical representation of the problem to be solved. The goal is to optimize the numerical solution, where optimization means maximization or minimization of linear equations. An ILP specification has two parts, an objective function and constraints. The general form of the objective function is:

$$\max : f(X_1, \ldots, X_n) := y_1 X_1 + \ldots + y_n X_n$$

The general form of the constraints is:

$$a_{i1} X_1 + a_{i2} X_2 + \ldots + a_{in} X_n \left(\begin{array}{c} \leq \\ = \\ \geq \end{array} \right) b_i,$$

with $i = 1, \ldots, m$

[1] There are also verbs such as "to give" where both resolution alternatives are allowed, given an appropriate context.

X_i are variables, y_i, b_i and a_{ij} are constants. The goal is to maximize (or minimize) a n-ary function f, which is defined as the sum of $y_i X_i$.

ILP as a scheme for global inference in NLP has been introduced by [16] and applied to various NLP tasks, including generation of coherent discourse [1], shallow dependency labeling [6] and semantic role labeling [15].

3 Binary ILP Models

Coreference resolution can be modeled as a binary classification task: two markables are or are not coreferent. Given n markables, numbered according to their occurrence in a text, $1..n$ (the markable indices), a binary relation $\mathcal{C}_{ij}$ with $i < j$ represents a classification decision: If $\mathcal{C}_{ij} = 1$, then the markables with position index i is the antecedent of markable j, which is the anaphor. If $\mathcal{C}_{ij} = 0$, the two markables are not coreferent. $\mathcal{C}_{ij}$ represents a candidate pair. Whether it is (set to) one or zero, depends on a number of constraints (e.g. do they agree) and the strength or weight that such a coupling receives according to an underlying statistical model. We rely on a machine learning (baseline) classifier to fix these weights of a candidate pair. Constraints can be formulated with (in)equalities, e.g. $\mathcal{C}_{ij} \leq \mathcal{C}_{ik}$. This is the ILP equivalent to implication from statement logic. It might be instructive to relate binary ILP modeling to statement logic and find mappings from logical connectives to their counterparts within the ILP framework. Such a bridge could ease the understanding of our formalization and help to evaluate the potential of ILP for NLP. The binary relation $\mathcal{C}_{ij}$ can be reinterpreted as a propositional variable: it can be true or false. Constraints corresponds to formulas, i.e. expressions combining propositional variables and logical connectives such as implication or disjunction. The most obvious difference between these two reasoning schemes is that in the case of ILP the inferences are driven by optimization. ILP is - in a sense - model building under the supervision of optimization.

In Fig. 1, we give a mapping from logic formulas to ILP equations (X_i are binary variables)

1.	$X_1 \veebar .. \veebar Xn$	$X_1 + .. + X_n = 1$
2.	$X_1 \vee .. \vee X_n$	$X_1 + .. + X_2 \geq 1$
3.	$X_1 \wedge .. \wedge X_n$	$X_1 + .. + X_n = n$
4.	$X_1 \rightarrow X_2$	$X_1 \leq X_2$
5.	$X_1 \leftrightarrow X_2$	$X_1 = X_2$

Fig. 1: *Statement Logic and ILP*

Exclusive OR (cf. line 1 in Fig. 1) requires that exactly one propositional variable is set to one, i.e. the sum of all variables must be one. Logical OR excludes the possibility that all variables are set zero, thus, the sum of all variables must be at least one. AND of n variables must sum up to n - setting each variable to one. Implication is false, if the antecedent is true, but the consequent is false. This means that the value of the antecedent is less or equal to the consequent. If the antecedent is one, then the consequent must also be one, otherwise the formula is not fulfilled. Equivalence of two variables means that they must be equal.

4 Transitivity

One crucial property of the anaphoric relation is transitivity. For three markables i, j, k to be coreferent it must hold that $\mathcal{C}_{ij} \wedge \mathcal{C}_{jk} \rightarrow \mathcal{C}_{ik}$, or, in a simpler notation: $X_1 \wedge X_2 \rightarrow X_3$. According to Fig. 1 line 4, $X_2 \rightarrow X_3$ corresponds to $X_2 \leq X_3$. Adding a further antecedent X_1 to the lefthand side of the (in)equality increases the amount at most by 1 (if $X_1 = 1$). Accordingly, we must add the amount of 1 on the righthand side - to keep the balance. So our transitivity statement becomes: $X_1 + X_2 \leq X_3 + 1$ (cf. also [3]).

The point here is, if X_1 and X_2 are set to one, then X_3 is forced also to be one by ILP. If only one antecedent is set to one, nothing can be deduced with respect to the consequent. And since the sum of the antecents are restricted to be less (or equal) to the consequent, the value of the consequent has no influence on the values of the antecedents.

Note that $X_1 \wedge X_2 \rightarrow X_3$ is only one incarnation of transitivity constraints on these three binary relations X_1, X_2, X_3. Also $X_3 \wedge X_1 \rightarrow X_2$ and $X_3 \wedge X_2 \rightarrow X_1$ are valid and must be generated to take full advantage of ILP's reasoning capabilities.

Coming back to our former notational conventions, the full definition of transitivity is:

$$\mathcal{C}_{ij} + \mathcal{C}_{jk} \leq \mathcal{C}_{ik} + 1, \quad \forall i, j, k \ (i < j < k) \tag{1}$$
$$\mathcal{C}_{ik} + \mathcal{C}_{jk} \leq \mathcal{C}_{ij} + 1, \quad \forall i, j, k \ (i < j < k) \tag{2}$$
$$\mathcal{C}_{ij} + \mathcal{C}_{ik} \leq \mathcal{C}_{jk} + 1, \quad \forall i, j, k \ (i < j < k) \tag{3}$$

In our model, transitivity is mainly used to propagate exclusiveness. It is the primary mechanism to enforce consistency on coreference sets. Exclusiveness stems from binding and agreement constraints. We first discuss our ILP model and then come back to transitivity as a mechanism that "globalizes" local exclusiveness constraints.

5 ILP's Objective Function

Our ILP model is straightforward[2]. The objective function introduces for each positive (and only for positive) classification decision of the baseline classifier a indicator variable, $\mathcal{C}_{ij}$, that is weighted by its corresponding probability, $\mathcal{P}_{ij}$.

The objective function is:

$$max: \sum_{(i,j) \in \oplus} \mathcal{P}_{ij} * \mathcal{C}_{ij} \tag{4}$$

$\oplus$ is the set of positively classified pairs, $\mathcal{P}_{ij}$ is the probability of such a pair used as a weight and $\mathcal{C}_{ij}$ is the indicator variable that eventually is set to zero or one. If it is set to one, then ILP has adopted the classification decision of the baseline classifier, otherwise, if set to zero, ILP has revised it. It is important to note that our model relies exclusively on positively classified pairs. This is due to the fact that binary classifiers (including our baseline classifier) are unaware of transitivity: as a consequence, a pair that is transitively implied by two positively classified pairs might - at the

[2] We use lp_solve, cf. http://lpsolve.sourceforge.net/.

same time and inconsistently- get a negative classification.

Given two positively classified pairs, (i, j) and (j, k). Although (i, k) transitively follows, the binary classifier often assigns a negative classification to such pairings (i, k). This is due to its "global blindness", but it is - from a local perspective - quite reasonable. Assume a proper name, i, at the beginning and a pronoun, k, at the end of a text. Both might be in the same coreference set (via a long chain of intermediary mentions), but there is no apriori or empirical reason that their direct linkage must form a good candidate pair as well. On the contrary, a personal pronoun hardly refers to the same referent throughout a (long) text, it shifts forth and back acting as a local variable .

Thus, positive classifications (of the baseline classifier) are better indicators of coreference than negative ones are indicative of exclusiveness: some of the negative classifications are - from a transitive (global) perspective - contradictory. A model that takes transitivity into account as our model does must not get confused by flaws inherent to lower level models that ignore transitivity. Therefore, we don't consider negative classifications. In the worst case, a coreference set stemming from the binary decisions of the baseline classifier could have more negative than positive links (a coreference set with 4 markables already might have the same number of positive and negative links). An ILP model that tries to optimize on the basis of all pairings might find itself in the uncomfortable position to destroy a coreference set (since it might have a negative sum, i.e. a negative addend). In same cases, this might be (accidentally) appropriate. However, the probabilistic model of the baseline classifier simply does not license this kind of reasoning, since it is beyond its scope.

6 Linguistic Constraints

Transitivity is a structural constraint as it defines how the (truth) values of indicator variables depend on each other. We have already introduced transitivity in section 3. Here we deal with linguistic constraints namely intra-sentential binding constraints. In our ILP model, binding constraints are used for exclusiveness restrictions. In the literature, various versions of a binding theory are being discussed. Often the coindexing of arguments (mostly pronominal or nonpronominal NPs) is restricted by a structural relation over phrase structure trees - the c-command. We give here a simple version of such a binding theory following [4]. Note, that there are also definitions of binding constraints in other syntactic frameworks such as dependency grammar (cf. the d-bind command in [18]).

C1 A reflexive pronoun must be coindexed with a c-commanding argument within the minimal NP or S that contains it.

C2 A nonreflexive pronoun must not be coindexed with a c-commanding NP within the minimal NP or S that contains it.

C3 A nonpronominal NP must not be coindexed with a c-commanding NP.

Only [C2] and [C3] define exclusiveness, we therefore discard [C1] from consideration (but see section 10). Moreover, since we can't rely on perfect parse trees (a statistical parser is being used in our experiments), we do not work with the c-command. Instead, we define a simple predicate, *clause_bound*, that most of the time correctly captures the restrictive function of the c-command. Two mentions, i and j, are *clause bound*, if they occur in the same (sub)clause, none of them is a reflexive or a possessive pronoun and they don't form an apposition. There are only 16 cases where this predicate produces false negatives. Some of these cases are country names reoccuring in the same clause as part of an adjectival phrase ("Russia$_i$ and Russian$_i$ people ..."). False negatives might also stem from clauses with predicative verbs ("He$_i$ is still prime minister$_i$").

- ILP version of [C2] and [C3]

$$\mathcal{C}_{ij} = 0, \quad \forall i, j \ (clause\_bound(i, j)) \tag{5}$$

Two markables i, j that are clause bound (in the sense defined above) are exclusive.

A possessive pronoun is exclusive to all markables in the (base) noun phrase it is contained in (e.g. "her$_i$ manager$_j$" with $i \neq j$), but might get coindexed with markables outside of such a local context ("Anne$_i$ talks to her$_i$ manager"). We define a predicate *np_bound* that is true of two markables i, j if they occur in the same (base) noun phrase. In general, two markables that *np bind* each other are exclusive. This is captured by the following constraint.

- Exclusiveness in local contexts

$$\mathcal{C}_{ij} = 0, \quad \forall i, j \ (np\_bound(i, j)) \tag{6}$$

A structural or technical constraint completes our ILP model: the definition of variables as being binary integer variable. Being binary is a constraint that must be modeled explicitely, while being an integer simply can be declared (by a keyword 'int').

- Variables are binary

$$\mathcal{C}_{ij} \in \{0, 1\}, \quad \forall i, j \ (i < j) \tag{7}$$

7 Some Properties of the Model

The predominant property of our model is that it leaves the decisions of the binary classifier unaltered as long as no inconsistencies arise. Only then, coreference sets are reconstructed.

Inconsistencies can be identified locally, namely if exclusiveness constraints are violated within a coreference set. For every two positively classified mentions i, j (occuring in the same clause), the exclusiveness constraints are checked. If a pair violates a constraint, the corresponding ILP indicator variable is explicitely set to zero, i.e. $c_{ij} = 0$. If these two mentions are not in the same coreference set (according to the baseline classifier), nothing happens - although their exclusiveness might serve as an additional restriction if a reorganization is triggered from other pairs.

On the other hand, an exclusive pair that is part of an inconsistent coreference set is a starting point for reorganization. Such an inconsistent coreference set might include other exclusive pairs, (k, l), this time only transitively given. Removing this kind of inconsistency is the task of the ILP reasoner. Pairs that do not (directly) violate an exclusiveness restriction thus are introduced into the ILP model simply as indicator variables without a predetermined value. Their values will get fixed as part of the global optimization and will adhere to the global constraints (i.e. transitively propagated exclusiveness). If c_{kl} is inconsistent, then the ILP program will assign it the value zero. Otherwise, it will receive the value one.

Although (some) violations can be found locally, simply removing one of the inconsistent mentions from the inconsistent coreferent set does not in any case solve the problem. Nor does applying a hard filter as part of the vector generation component of the binary classifier prevent the inconsistent coreferent set from being generated (as we have argued).

Given a binary classifier that relies on our binding constraints as a hard filter. That is, if two mentions i, j are exclusive, the pair (i, j) won't be generated. Suppose that no (locally detectable) binding constraints are violated for (h, i), (i, k) and (j, k) (i.e. they agree and are in different clauses) and that the classifier classifies them as coreferent. This gives rise to the inconsistent coreference set {h,i,j,k}. However, we can't blame the classifier for this. Neither does it know of the exclusiveness of i, j, nor could it use this knowledge, since its classification trials are independent of each other. Transitivity is beyond the horizon of such approaches - they are "globally blind". Moreover, (simply) removing, let's say, i from {h,i,j,k} does not yield a consistent coreference set, since $c_{hi} = 1$ and $c_{ij} = 0$ imply $c_{hj} = 0$[3].

What our ILP model does in such cases: it splits the inconsistent coreference set into (at least) two consistent sets. Sometimes, e.g. if a mention is exclusive to all other mentions in the original set, or if the set consists only of two (the inconsistent) mentions, one or both of them might become non-anaphoric (i.e. left unattached). If they are, however, true mentions according to the gold standard, then recall drops. Most of the time, however, ILP preserves or boosts recall (see section 9).

8 Performance Aspects

Although defining an ILP model is concise, applying it to real problems is not, since every formula must be extensionalized. Depending on the number of positively classified pairs, such a extensionalized collection of formula statements can get rather large. Especially the compilation of transitivity statements produces a vast number of lines. Most of the time, ILP comes up very quickly with a solution even if given thousands of particular constraints. However, it is impossible to extensionalize transitivity for longer texts (e.g. whole books). Fortunately, there is a natural division, i.e. splitting texts in paragraphs or chapters. Coreference resolution could then be done incrementally, preserving found solutions by defining segmentation overlaps (i.e. a window moving over the text).

In our experiments, we haven't integrated a model of text segmentation or text tiling for coreference resolution. Wherever possible, we run ILP on the whole text. In those rare cases where ILP didn't stop, we used a predefined window[4]. While moving the window over the text, former classification decisions are simply taken along, such that the final window comprises the accumulated ILP assignments. Clearly, the restrictive power of transitivity statements is weakened this way, since transitivity only applies within the window. However, the only class of mentions that coherently can bridge long distances are matching proper names. They are, however, hardly involved in the propagation of inconsistencies (which to detect and resolve is the primary objective of transitivity in our model).

9 Empirical Evaluation

As a baseline system we use a reimplementation of the Soon coreference classifier (cf. [17]). The baseline system features and its performance with respect to the MUC-6, MUC-7 and ACE data are described in [14][5]. The ACE data [10] is used as a corpus. Since our model does not need a training phase, no splitting of the texts (sections Newswire (NWIRE) and Broadcast News (BNEWS)) was necessary.

We report in the following tables the MUC score [19], but we also have measured the system's performance with a metric called ECM-F, introduced by [7] (the Bell tree approach). ECM-F is an acronym for entity-constrained mention F-measure. It first aligns the system entities (i.e. the found coreference sets) with the reference entities (i.e. the gold standard coreference sets). This is done in a way such that the number of common mentions is maximized. However, each system entity is constrained to align with at most one entity from the reference set (and vice versa). ECM is a very tough metric that has its shortcomings, but it is sensitive to the primary (splitting and reordering) effect of our ILP model. As we will argue, it is better suited than the MUC score.

Consider as an illustration the schematic example from Fig. 2. Here [[m1,m2,m3],[m4,m5]] is the gold standard (two coreference sets, 5 mentions), the baseline classifier produces, say, [[m1,m2,m3,m4,m5]] (one coreference set with all 5 mentions in it). Assume that m3 and m4 are exclusive. In this case, ILP might produce [[m2,m3],[m1,m4,m5]], for example. This would be a reasonable splitting. However, the MUC score prefers the baseline classifier's partition (0.857 F-measure) over the ILP results (0.667). Quite contrary, the ECM metric prefers ILP's solution (0.8 versus 0.6)[6]. As discussed in [2], every metric has

[3] $(\mathcal{C}_{hi} + \mathcal{C}_{hj} \leq \mathcal{C}_{ij} + 1) \wedge (\mathcal{C}_{ij} = 0) \rightarrow (\mathcal{C}_{hi} + \mathcal{C}_{hj} \leq 1)$. Given $\mathcal{C}_{hi} = 1, \mathcal{C}_{hj} = 0$ follows.

[4] This could be easily automated with the time-out mechanism of lp_solve.

[5] I would like to thank Simone P. Ponzetto and Michael Strube for making me available the results of their Soon reimplementation applied to the ACE texts.

[6] ECM evaluation: 5 true mentions, ILP postulates 5 mentions, corefset1 and corefset2 of ILP and the gold standard align, respectively. 4 mentions are in common, thus recall and precision are 4/5.

reference	corefset1 [m1,m2,m3]	corefset2 [m4,m5]	MUC P	R	F	ECM P	R	F
classifier	[m1,m2,m3,m4,m5]		3/4	3/3	0.857	3/5	3/5	0.6
ILP	[m2,m3]	[m1,m4,m5]	2/3	2/3	0.667	4/5	4/5	0.8
ILP effect					drop			boost

Fig. 2: *Illustration: MUC score compared to ECM score*

its strengths and shortcomings - it sheds light on certain aspects and hide others. It seems reasonable to choose at least one suitable measure for the problem at hand.

Given the MUC score, in our schematic example ILP even reduces the performance of the baseline classifier (cf. ILP effect "drop" from Fig. 2), while with ECM ILP boosts performance ("boost"). Please note that in our real experiments, even the MUC score admits ILP an increase over the baseline classifier, although a considerably smaller one than the one of the more suitable ECM (cf. Fig. 3).

Before we come to discuss our experimental results, we would like to stress another point. Recent work on coreference resolution (not only with the ACE texts but more generally) often works with true mentions (i.e. only markables that are in the gold standard). This is a considerable simplification, since the classification of a markable as being a true mention itself is an error prone task. Performance thus considerably drops, if one returns to a realistic scenario where all markables are to be related. We don't want to criticize the "perfect settings" - the reason why we run our experiments in a realistic setting is not purism[7]. Remember that our model becomes active only in those situations where coreference sets are inconsistent. With perfect data (BNEWS and NWIRE) only a few exclusiveness violations take place (namely 34 violations as compared to 180 given the realistic data). Even with the perfect data some improvements were achieved (2% ECM-F-measure, no improvement according to MUC score). However, our model seems to contribute more to the realistic setting than to the perfect.

	NWIRE			BNEWS		
	P	R	F	P	R	F
BL_{MUC}	46.6	62.6	53.4	55.6	60.9	58.1
ILP_{MUC}	56.1	56.5	**56.3**	63.4	56.1	**59.5**
BL_{ECM}	45.6	50.5	47.0	56.0	44.1	46.8
ILP_{ECM}	53.2	54.4	**53.3**	60.1	44.9	**49.3**

Fig. 3: *MUC score compared to ECM score*

Fig. 3 shows the empirical results. We give the MUC scores for each text collection as well as the ECM results. 'BL' is the Soon baseline classifier, 'ILP' our ILP model. According to the MUC score, ILP pushes NWIRE results by 2.9% and BNEWS by 1.4%. Note that precision rises significantly but at the same time recall drops. This is due to the bias the MUC score obeys to (as discussed above). With ECM evaluation the situation is better, NWIRE is pushed by 6.3% and BNEWS by 2.5%, so we have an improvement of 4.4% for the whole collection.

With NWIRE, there are 114 direct exclusiveness violations, while in BNEWS there are 66 violations. 'direct' means 'locally observable', the actual number of violations (considering also transitivity) is higher. Instead of counting these cases, we measured the impact of transitivity more directly. Fig. 4 gives the results of this experiment, where we removed transitivity from the ILP model. Thus, ILP selects (and optimizes) new coreference sets only according to the given weights (adhering still to local exclusiveness).

	NWIRE			BNEWS		
	P	R	F	P	R	F
BL_{ECM}	45.6	50.5	47.0	56.0	44.1	46.8
ILP_{trans}	53.2	54.4	53.3	60.1	44.9	49.3
ILP_{no_trans}	50.5	53.0	**51.0**	59.8	45.5	**49.5**

Fig. 4: *Does transitivity matter (ECM score)?*

With the NWIRE texts (112 violations), transitivity contributes 35% to the improvement (47.0% was the baseline, 53.3% the effect of full ILP and 51.0% has been achieved without transitivity). There is no effect with the BNEWS texts (66 violations). Whether transitivity helps might correlate with the number of violations or even the cardinality of the coreference sets (larger sets might profit from transitivity propagation). Further experiment are necessary to fix this.

10 Towards a Full ILP Model

There are various conceivable ILP models for coreference resolution. We have defined one that can be interpreted as a repair model. It removes inconsistency and relies otherwise on the baseline classifier's decisions. Such a model boosts first and foremost precision but might also - as a side effect - increase recall.

However, ILP's role could be more active. It could try to raise pairs from false negatives to true positives. The detection of false negatives, however, is a tricky business. The probabilities of the baseline classifier could not help here (as they have caused the false negatives). So only further linguistic constraints could help. That is, the more normative restrictions and predetermined indicator values are given to ILP the better it will perform. Anaphoricity detection in the sense of [13] might play a role, even a model of bridging could narrow down the freedom of coreference decisions (if it is known that two mentions are in a bridging relation, they are exclusive and each of their coreferent mentions are as well).

In the current model, only positively classified pairs are being considered. To increase recall, also neg-

[7] Although with the perfect setting, a very simple strategy that sets all linkages positive receives (with ACE texts) a very high MUC score (F-measure around 80 %), cf. also [2].

atively classified pairs would have to be considered. Negatively classified means a probability < 0.5. One could give these pairs a negative weight. If the linguistic constraints are strong enough, such pairs might get raised in spite of their negative weight to become part of a coreference set. Alternatively, the threshold - separating negative from positive pairs could be lowered to, say, 0.4. This way, a number of false negatives automatically would be raised to be true positives, but also a number of true negatives would become false positives. Again, linguistic constraints could help, this time suppressing false positives.

What could these additional constraints look like. Here are some them, starting with the very fundamental agreement constraint.

- Agreement constraint:

$$\mathcal{C}_{ij} = 0, \quad \forall i, j \ (\neg agree(i, j)) \tag{8}$$

Two markables that do not agree are exclusive. The definition of agreement depends on the type of the markables (e.g. two personal pronouns must agree in number, person and gender).

Another constraint is the ILP equivalent of the binding constraint [C1] (cf. section 6).

- Boundness of a reflexive pronoun:

$$\sum_{k=s}^{i-1} \mathcal{C}_{ki} + \sum_{j=i+1}^{e} \mathcal{C}_{ij} \geq 1 \tag{9}$$

where i is the pronoun and s (e) the start (end) index of the first (last) mention of the sentence in which i occurs.[8].

Or consider nominal anaphora. It is hardly the case that a coreference set comprises more than two or three "real" nominal anaphora, that is, reference by two or three different synonyms or super-concepts. Something like "Hanna .. my beloved sister .. the best comrade of my childhood .. this lucky girl" is clearly possible, but not very frequent, at least in newspaper texts. Provided this, or given any other empirically fixed upper bound of such references, one could restrict the number of non-matching NPs within a coreference set to, say, at most three different non-pronominal NPs. To achieve this, different types of indicator variables need to be introduced, including one for pairs of non-pronominal NPs.

Also, coreference sets that are built exclusively from pronouns are not valid. At least one non-pronominal mention seems to be required, otherwise the coreference set has no referential anchor (a definite description, a proper name, something that help to identify the real world object being refered to). Again, the introduction of a separate type of indicator variable would allow to define a corresponding restriction.

Another example: some constructions indicate that a NP is a nominal anaphor. For example, NPs with a attributive demonstrative pronoun such as "this" as in "this masterpiece" (sometimes such a reference is deictic, but rarely in written language). We can prevent such markables from being interpreted non-anaphorically by introducing a non-anaphoricity indicator variable and setting it to zero.

We believe that - given a number of such linguistic and heuristic constraints, ILP could even be more successfully boosting the empirical performance of traditional coreference resolution system based on binary classifiers.

11 Related Work

ILP as a tool to utilize the output of an underlying classifier to come to a consistent solution has been used so far only in few approaches (e.g. [1], [6], [8]). The architecture of all these systems is very similar (including ours), it more or less follows the design principles introduced in [16].

There is one recent approach to coreference resolution with ILP (see [5]). The paper is not yet published but will be presented at the NAACL 2007[9]. The most striking differences to our approach are: their ILP model does not integrate transitivity, it does not integrate binding constraints and uses perfect ACE data (only true mentions). Moreover, all mention pairs are combined and integrated into the objective function, whereas in our model only the positively classified pairs are being used. The authors discuss two models, the difference between them is the integration of indicator variables for anaphoricity. In our model, all positively classified instances are interpreted as anaphoric, so we don't need a separate indicator variable. Given these numerous differences, it is interesting to see that the impact of ILP in both approaches is similar, that is about 5%. However, the reasons for the improvement are quite different, since only our model operates with exclusiveness constraints and transitivity. The authors attribute their improvement to ILP's ability to globally optimize. At least with our baseline classifier this won't work (we have tried to replicate their results, but failed to do so). After all, the two approaches are distinct enough to co-exist: our model strives to boost performance via linguistically motivated constraints, while their models seem to profit from a better baseline system and a somewhat fuzzy notion of the benefits of "global optimization".

From those approaches that try to overcome the fixation on a binary coreference resolution (e.g. [9], [12] [20]) the Bell tree approach [7] is most important to be compared with. Here candidate coreference sets are being pursued in a (n-) best-first search by constructing a Bell tree (a tree with branches according to the Bell number with can be used to quantify the number of coreference sets given n markables). [7] do not integrate binding constraints, nor do they formalize transitivity within their model. The integration of a markable into a coreference set depends on it's probability with respect to the so-called active (i.e. last) element of the set. One difference to our model is

[8] In very rare cases, a reflexive pronoun is not bound in the subclause it occurs (e.g. German "Sich$_{refl}$ waschen$_{verb}$ hilft$_{verb}$.").

[9] Note that - at the time of writing this submission - we only had an unofficial "Google cached" HTML based version of that paper.

the representation of coreference sets. In our formalization, coreference sets are given intensionally (transitivity together with constraints), while in the Bell tree they are explicitly maintained, forcing the systems to prune (on performance grounds). Pruning of coreference sets, however, - if based on numerical measures only - is a local decision. In our approach, coreference sets are constructed simultaneously under the regiment of global optimization.

12 Conclusions

We have introduced a constraint-based model for coreference resolution within the framework of ILP. To the best of our knowledge, it is the first (empirically based) coreference resolution system that fully takes transitivity of the anaphoric relation into account. We have demonstrated that local linguistic constraints (intra-sentential binding constraints that give rise to exclusiveness of mentions) become globally operative via transitivity. Violated restrictions trigger the recomputation of coreference sets, transitivity guarantees consistency. If no violations occur, our model leaves the coreference sets of the baseline classifiers as they are.

We have empirically demonstrated that our system - especially if applied in a realistic setting - boosts the results of a baseline classifier considerably. We believe that a tighter coupling of empirical and theoretical knowledge in such (still numerical, but also normative) models is a step towards better NLP models. Future work will focus on the definition of a full ILP model of coreference. Such a model no longer only corrects inconsistent coreference sets, but autonomously generates coreference sets in the first place. In order for such an approach to work, more and tighter linguistic constraints are necessary.

Acknowledgment. I would like to thank Simone P. Ponzetto and Michael Strube for their support, be it with data or advice.

References

[1] E. Althaus, N. Karamanis, and A. Koller. Computing locally coherent discourses. In *Proc. of the ACL*. 2004.

[2] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *Proc. of the Linguistic Coreference Workshop at the 1st LREC*, pages 563–566. 1998.

[3] R. Barzilay and M. Lapata. Aggregation via set partitioning for natural language generation. In *Proc. of the HLT-NAACL*, pages 359–366. 2006.

[4] G. Chierchia and S. McConnell-Ginet. *Meaning and Grammar*. Cambridge, MA: MIT Press, 2001.

[5] P. Denis and J. Baldridge. Global, joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. of the NAACL(TO APPEAR)*. 2007.

[6] M. Klenner. Shallow dependency labeling. In *Proc. of the ACL, Poster (TO APPEAR)*. 2007.

[7] X. Luo, A. Ittycheriah, H.Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*, pages 135–142. 2004.

[8] T. Marciniak and M. Strube. Beyond the pipeline: Discrete optimization in NLP. In *Proc. of the CoNLL*. 2005.

[9] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems (NIPS)*. 2004.

[10] A. Mitchell, S. Strassel, M. Przybocki, J. Davis, G. Doddington, R. Grishman, A. Meyers, A. Brunstain, L. Ferro, and B. Sundheim. TIDES extraction (ACE 2003 multilingual training data). In *LDC2004T09, Philadelphia, Penn.: Linguistic Data Consortium*. 2003.

[11] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. New York: Wiley, 1999.

[12] V. Ng. Machine learning for coreference resolution: From local classification to global ranking. In *Proc. of the ACL*. 2005.

[13] V. Ng and C. Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proc. of the COLING*. 2002.

[14] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of the HLT-NAACL*, pages 192–199. 2006.

[15] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Semantic role labeling via integer linear programming inference. In *Proc. of the COLING*. 2004.

[16] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *Proc. of the CoNLL*. 2004.

[17] W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[18] M. Strube and U. Hahn. ParseTalk about sentence- and text-level anaphora. In *Proc. of EACL*, pages 237– 244. 1995.

[19] M. Vilain, J. Burger, J. Aberdeen, D. Conolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proc. of the 6th MUC*, pages 45–52. 1995.

[20] X. Yang, J. Su, and C. L. Tan. A twin-candidate model of coreference resolution with non-anaphor identification capability. In *Natural Language Processing - IJCNLP 2005, Second International Joint Conference*, pages 719–730. 2005.